ETH zürich

Meta-Learning

DRL Seminar

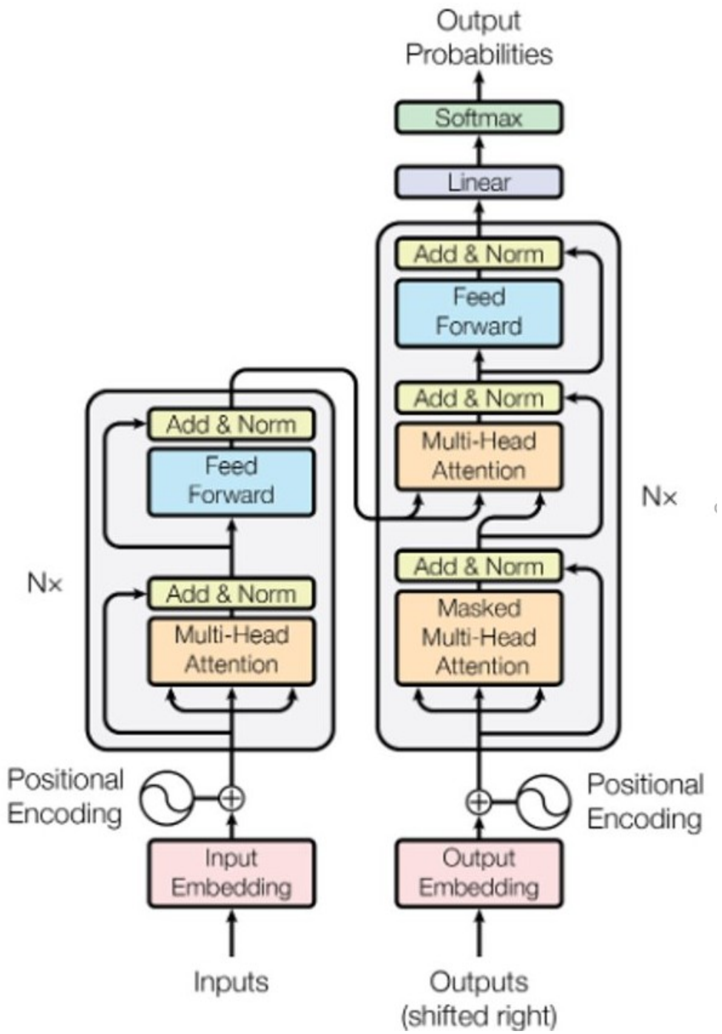
Philippe Blatter

Overview

- Introduction to Meta-Learning
- Model-Agnostic Meta-Learning (MAML)
- Optimization-based approaches
- Meta-Learning in RL

Supervised Learning Paradigm

- Large datasets
- Large models
- Long training time



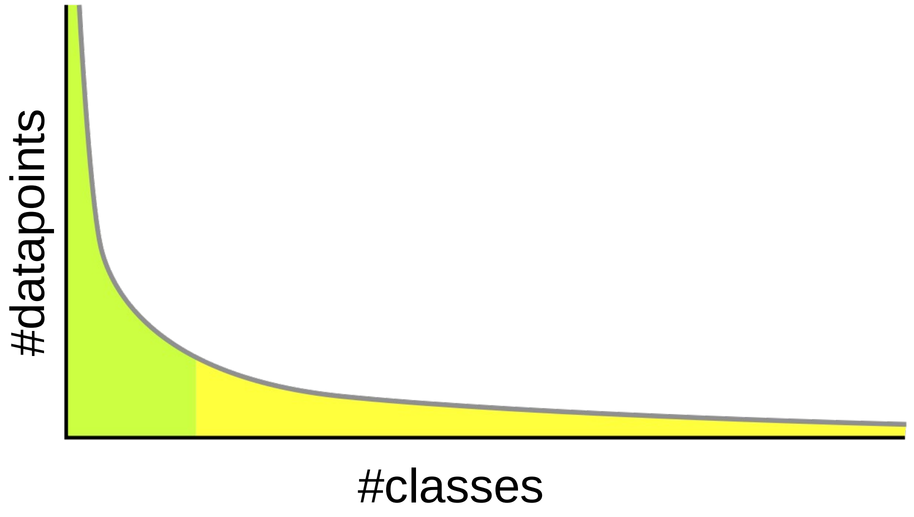
Transformer
([1] Vaswani et al. 2017)

Possible Problems

Large datasets might not be available

General-purpose AI

Long-tailed data



[2] Finn et al. 2017

Example

2-way

3 shots

Braque

Cezanne

Braque or Cezanne?

[2] Finn et al. 2017

Can we learn to learn?

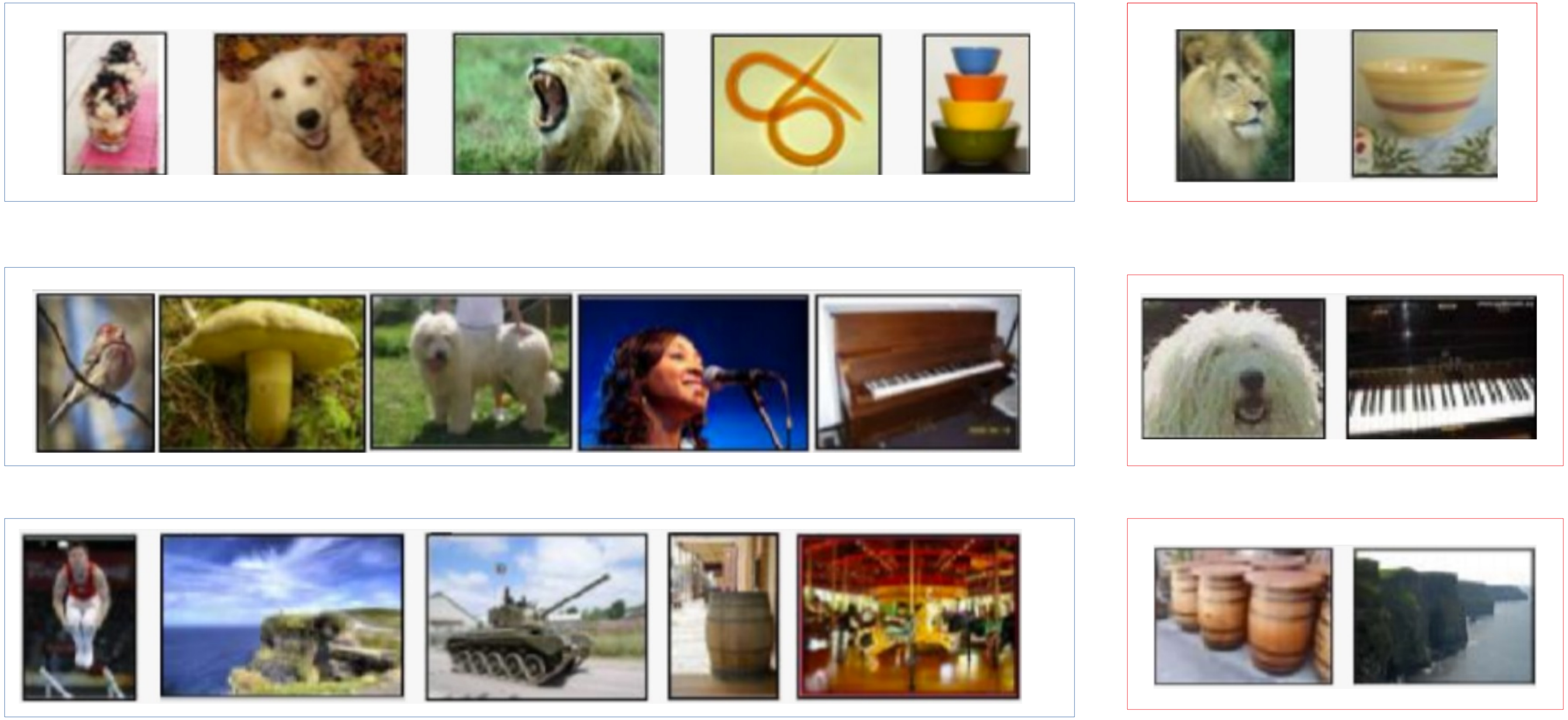
[2] Finn et al. 2017

Problem Setting

D^{tr} D^{ts}

D

$D_{\text{meta-train}}$ D_1 D_2



...

[2] Finn et al. 2017

Problem Setting

Supervised learning:

$$\arg\max_{\phi} \log p(\phi|D)$$

$$D = \{(x_1, y_1), \dots, (x_k, y_k)\}$$

Meta-learning:

$$D_{\text{meta-train}} = \{D_1, \dots, D_n\}$$

$$\arg\max_{\phi} \log p(\phi|D, D_{\text{meta-train}})$$

$$D_i = \{(x_1^i, y_1^i), \dots, (x_k^i, y_k^i)\}$$

[2] Finn et al. 2017

Meta-Learning Terminology

use $\theta^\star$ find $\phi^\star$

[2] Finn et al. 2017

Meta-Learning Problem

$$\theta^\star = \arg\max_\theta \log p(\theta | D_{\text{meta-train}})$$

meta-learning

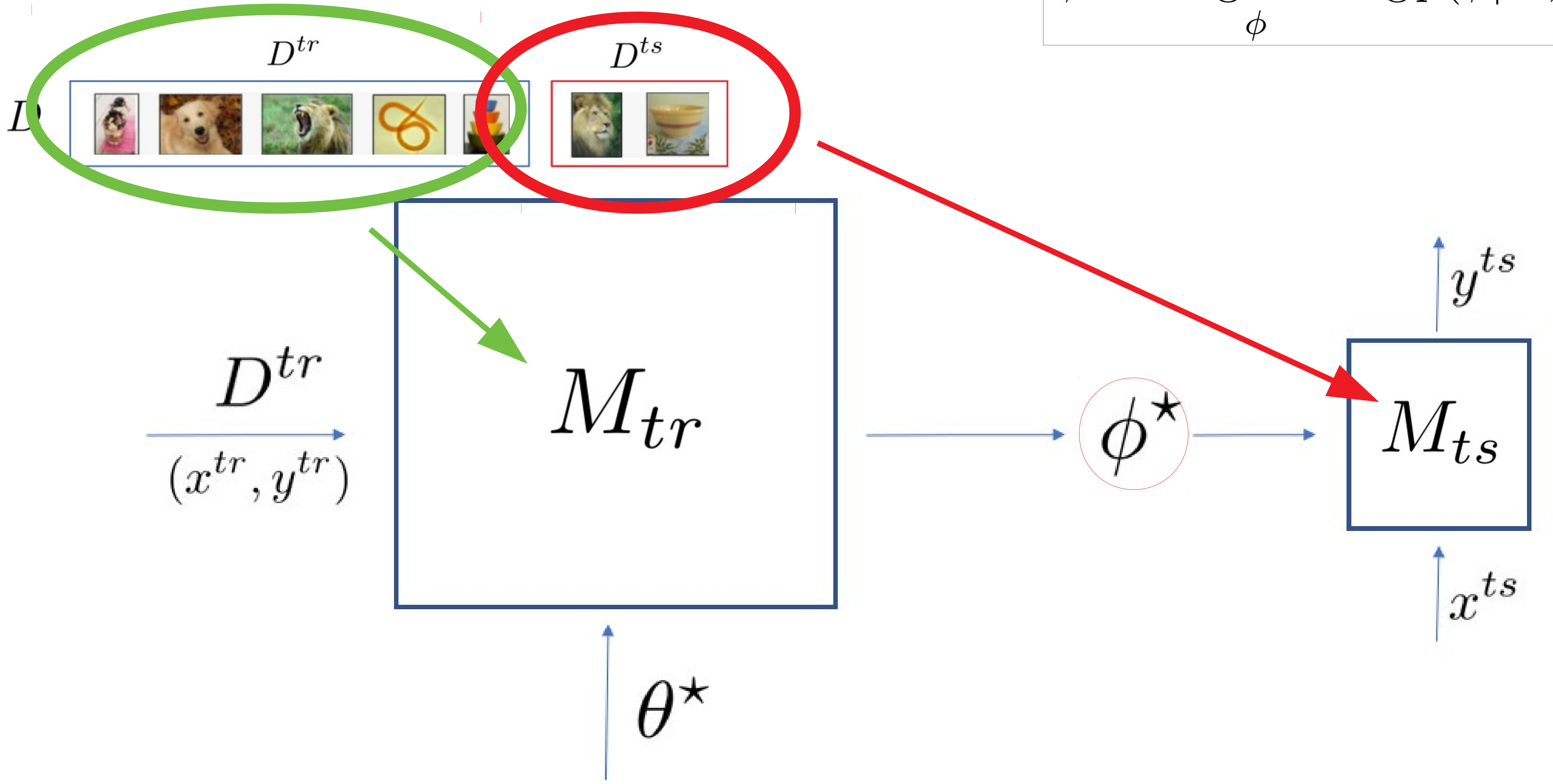
$$\phi^\star = \arg\max_\phi \log p(\phi | D, D_{\text{meta-train}}) = \arg\max_\phi \log p(\phi | D, \theta^\star)$$

adaptation

[2] Finn et al. 2017

(Meta) Test-Time

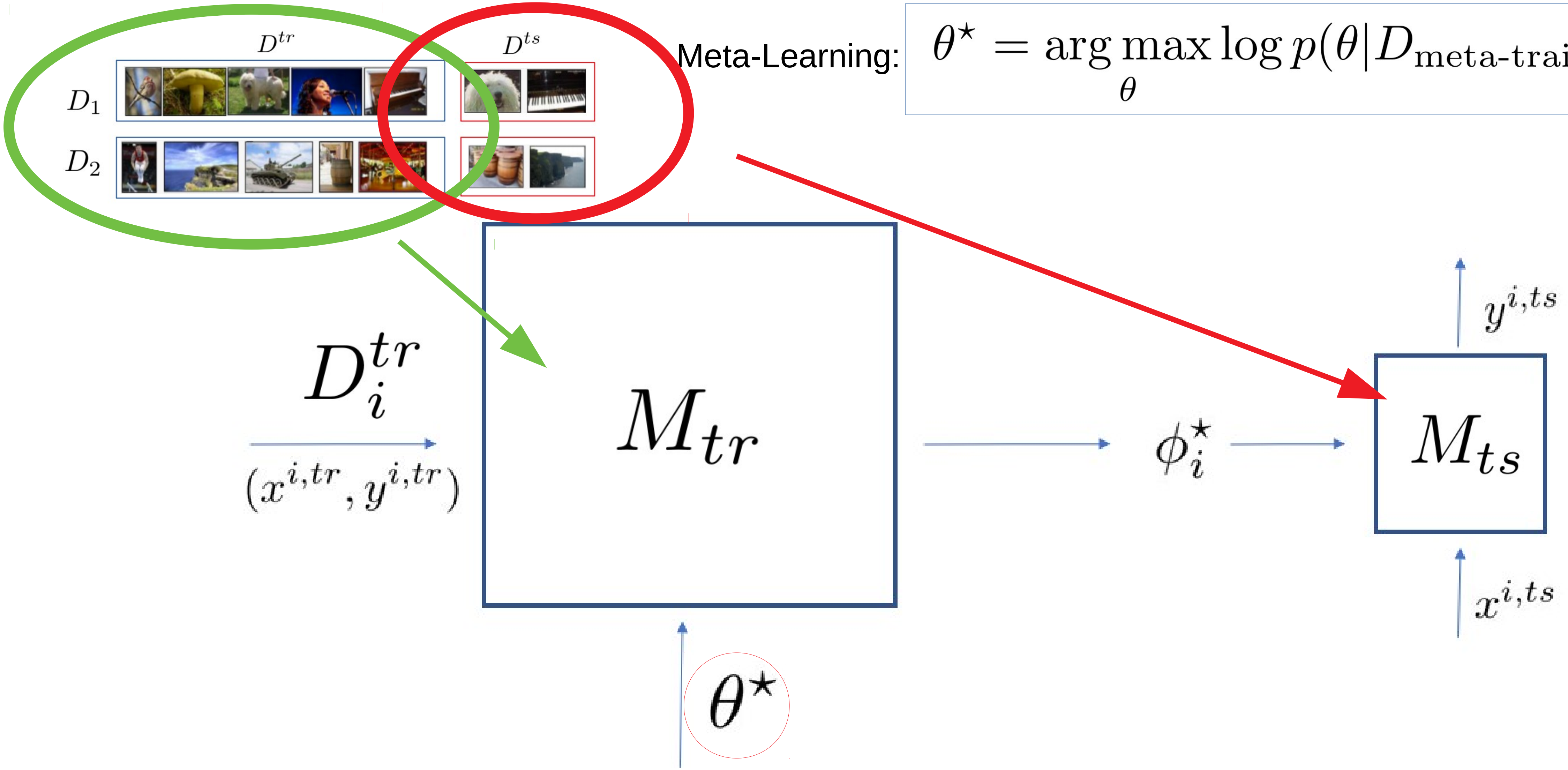
Adaptation: $\phi^\star = \arg\max_{\phi} \log p(\phi|D, \theta^\star)$



[2] Finn et al. 2017

(Meta) Training-Time

Meta-Learning: $$\theta^\star = \arg\max_\theta \log p(\theta | D_{\text{meta-train}})$$



[2] Finn et al. 2017

Complete Meta-Learning Problem

Meta-learning: $$\boxed{\theta^\star = \arg\max_\theta \log p(\theta | D_{\text{meta-train}})}$$

Adaptation: $$\boxed{\phi^\star = \arg\max_\phi \log p(\phi | D, \theta^\star)}$$

Learn θ such that $\phi_i = f_\theta(D_i^{tr})$ is good for D_i^{ts} for all tasks i

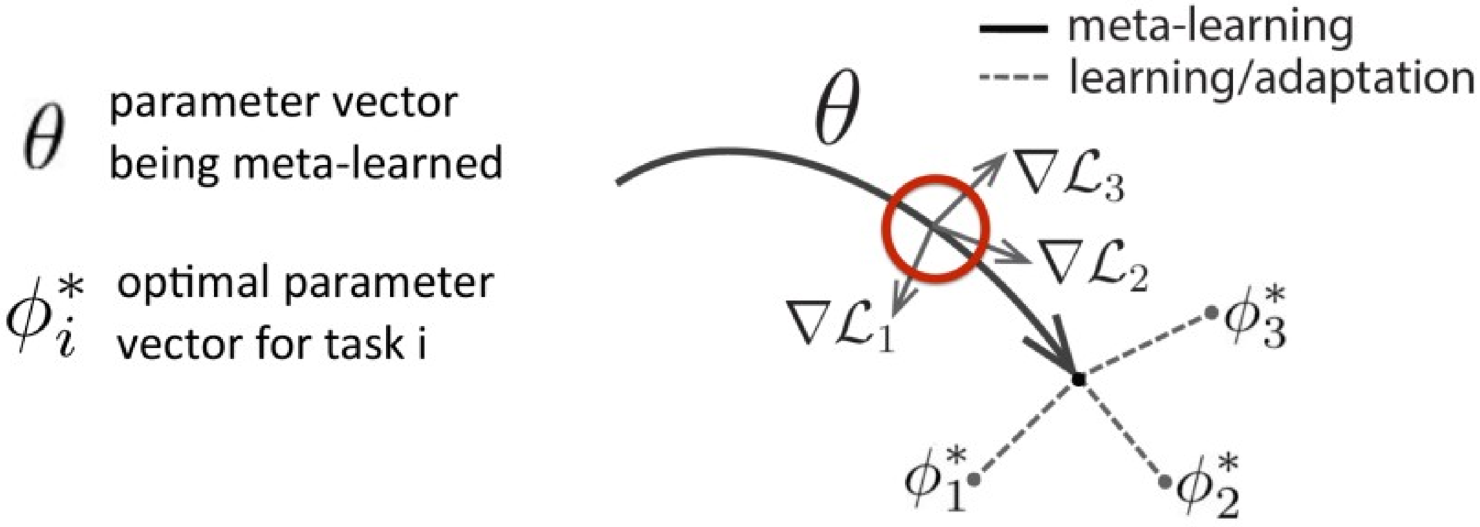
$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} \log p(\phi_i | D_i^{ts})$$

$$\text{where } \phi_i = f_\theta(D_i^{tr})$$

[2] Finn et al. 2017

Model-Agnostic Meta-Learning (MAML)

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$



Model-**A**gnostic **M**eta-**L**earning

[2] Finn et al. 2017

Model-Agnostic Meta-Learning (MAML)

- *"In our approach, the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task."*

pre-trained parameters

Fine-tuning [test-time]

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\mathrm{tr}})$$

training data for new task

Meta-learning

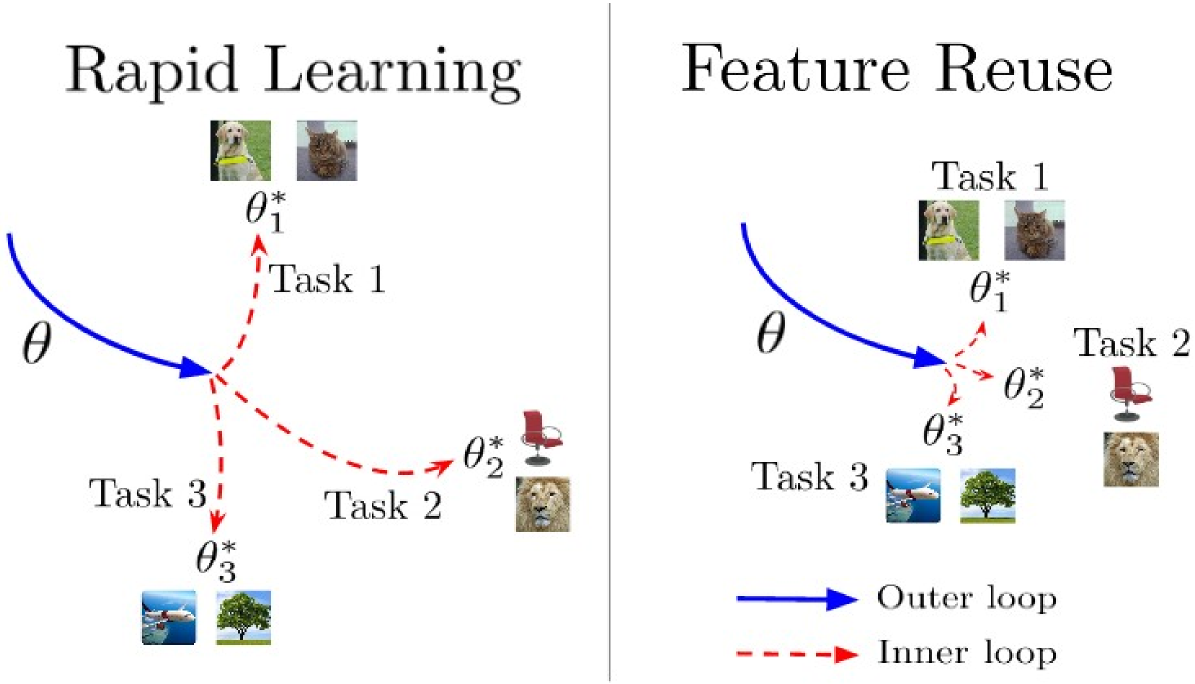
$$\min_\theta \sum_{\text{task } i} \mathcal{L}(\underbrace{\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\mathrm{tr}})}_{\phi_i}, \mathcal{D}_i^{\mathrm{ts}})$$

[2] Finn et al. 2017

Understanding the Effectiveness of MAML

- Rapid Learning: large representational changes occur during adaptation to new task

- Feature Reuse: Meta-initialization already contains highly useful features that can be reused for new tasks

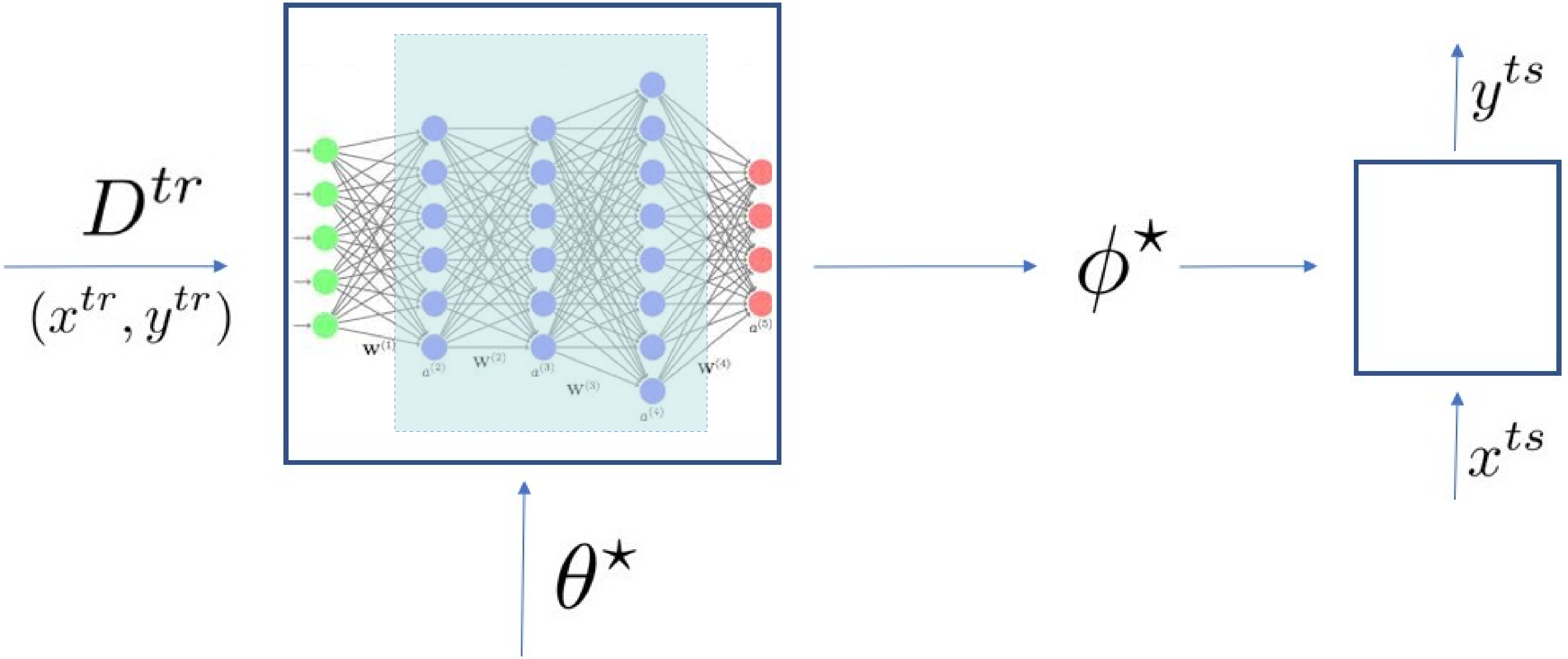


[3] Raghu et al. 2019

Freezing Layer Representations

Performance hardly changes.

-> Feature Reuse

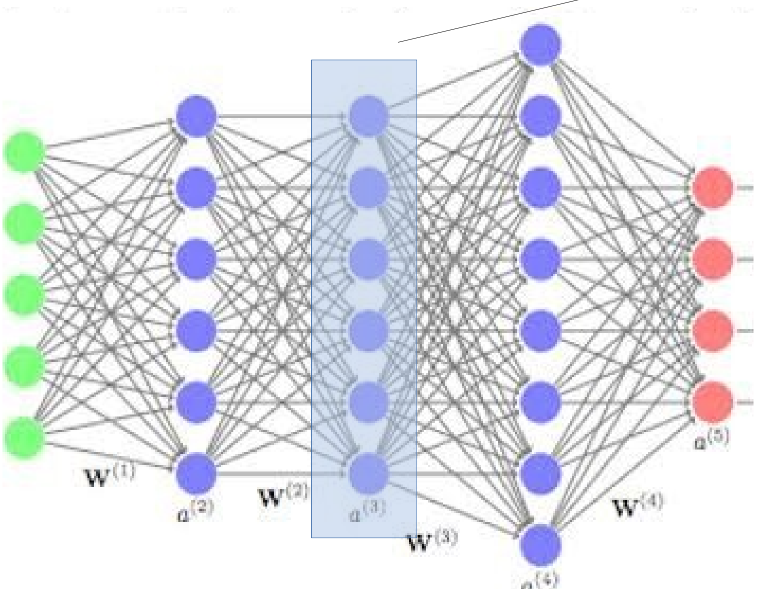


[3] Raghu et al. 2020

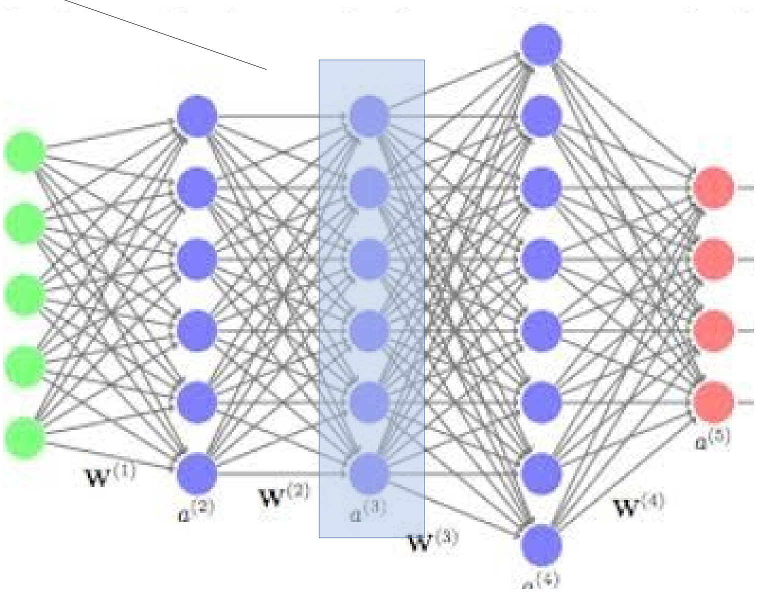
Representational Similarity Experiments

- Measure changes in the latent representations learned by the NN during apaptation using Canonical Correlation Analsysis (CCA)

CCA(L1,L2)



before adaptation



after adaptation

Highly similar representations in the body of the network

-> No functional change

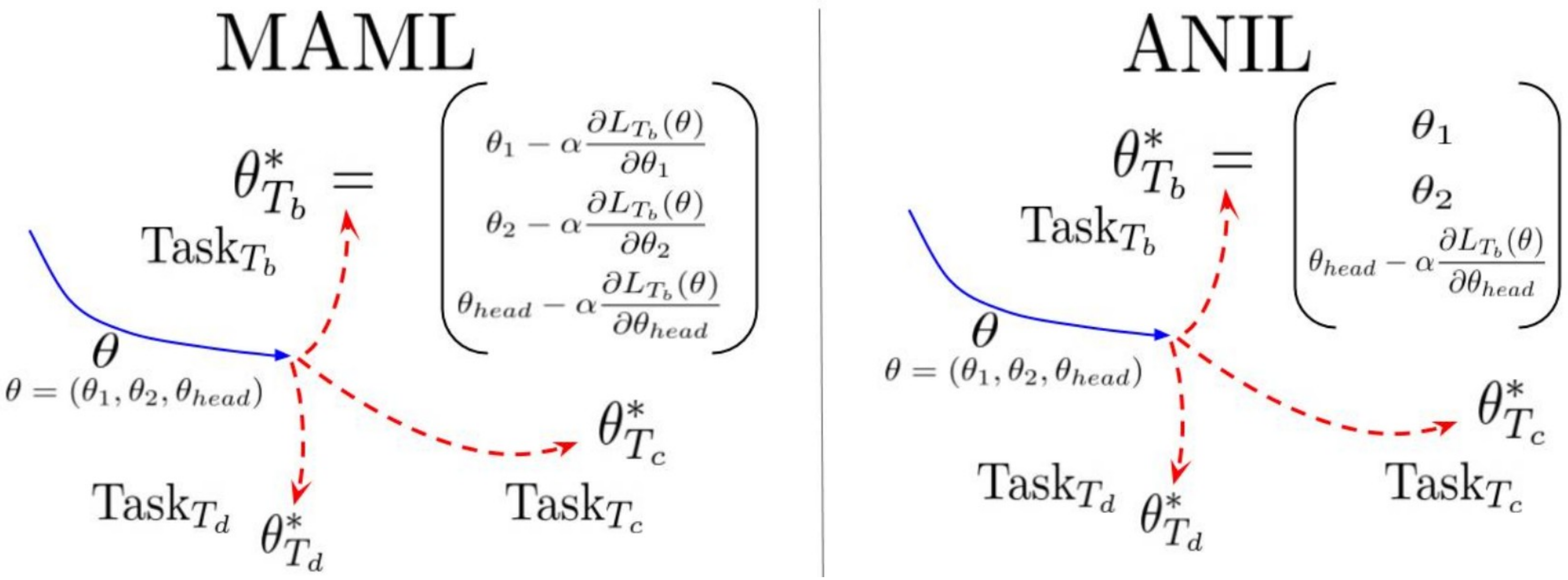
-> No rapid learning

[3] Raghu et al. 2019

ANIL Algorithm: Almost no Inner Loop (Adaptation)

Meta-learning $\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\underbrace{\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})}_{\phi_i}, \mathcal{D}_i^{\text{ts}})$

- Similar Performance to MAML



[3] Raghu et al. 2019

Learning to learn by gradient descent by gradient descent

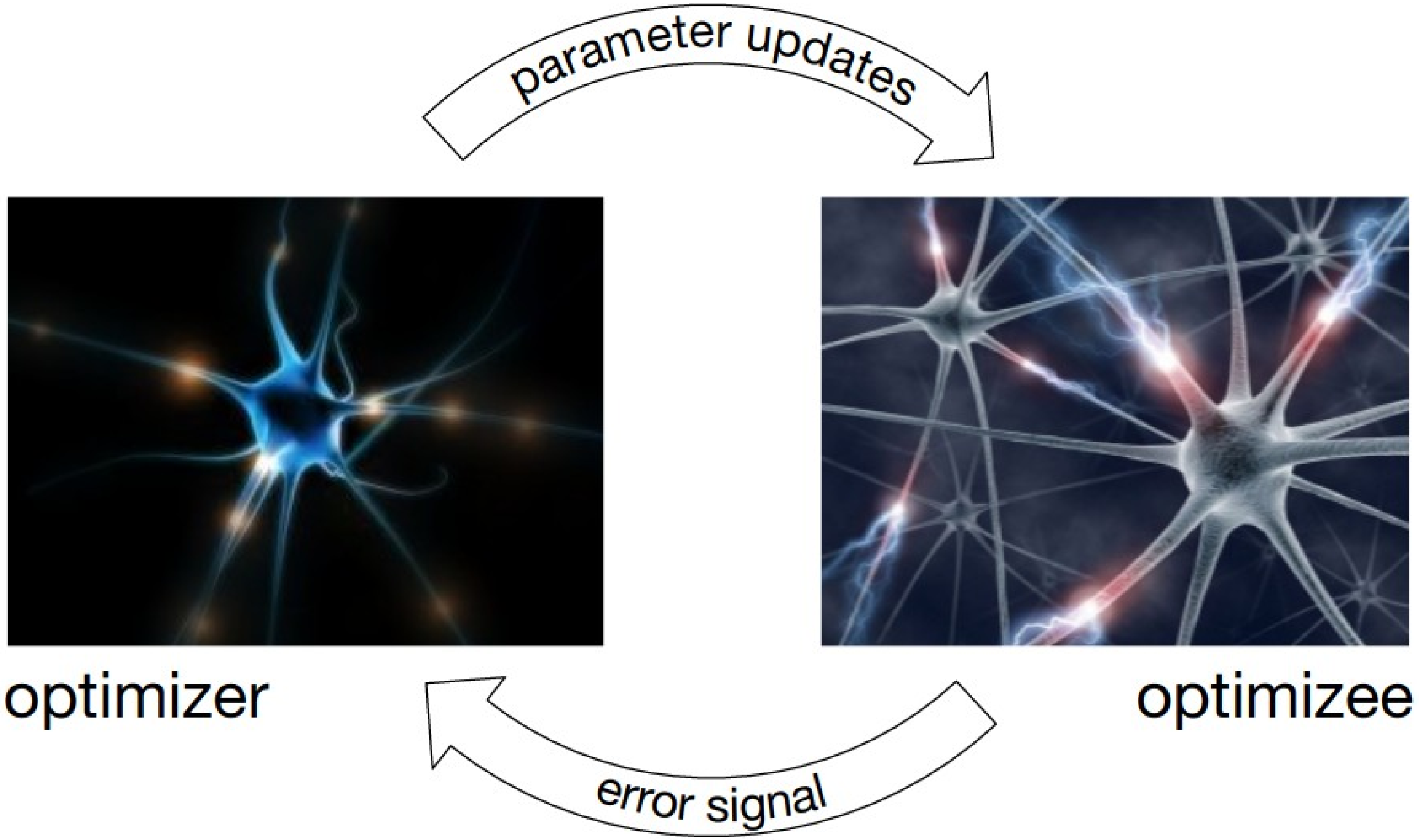
hand-designed features → learned features

hand-designed optimization algorithms → learned optimization algorithms

“Casting algorithm design as a learning problem”

[4] Andrychowicz et al. 2016

Learning to learn by gradient descent by gradient descent



[4] Andrychowicz et al. 2016

Learning to learn by gradient descent by gradient descent

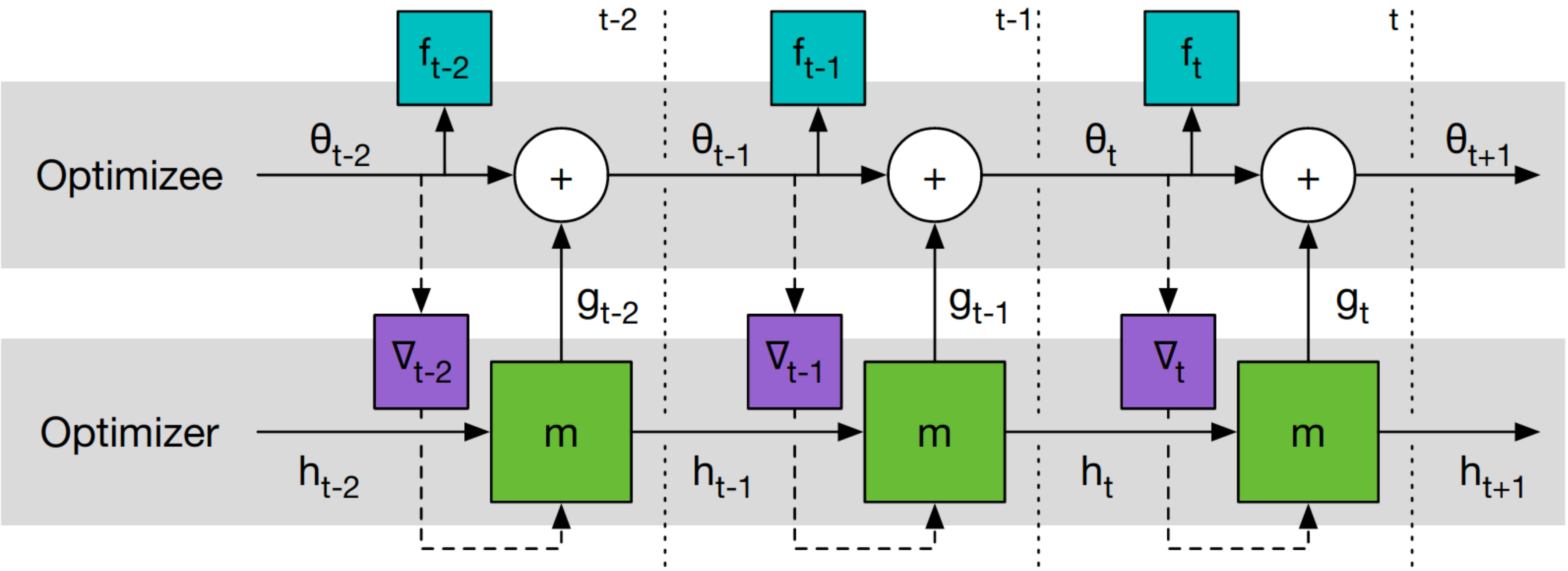
hand-designed optimization algorithms → learned optimization algorithms

"Casting algorithm design as a learning problem"

$$\theta_{t+1} = \theta_t - \alpha_t \nabla f(\theta_t) \quad \Rightarrow \quad \theta_{t+1} = \theta_t + g_t(\nabla f(\theta_t), \phi)$$

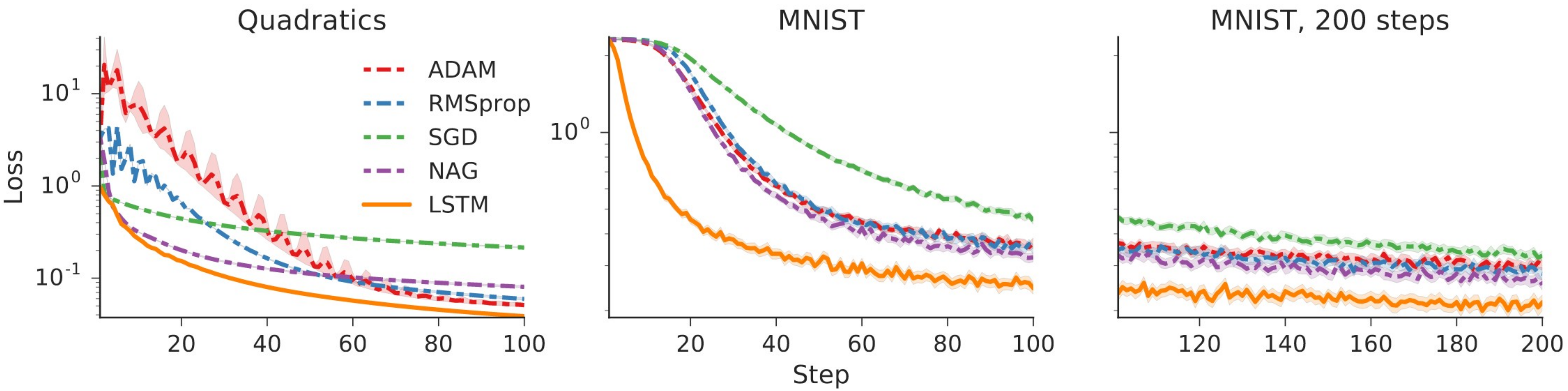
[4] Andrychowicz et al. 2016

Learning to learn by gradient descent by gradient descent



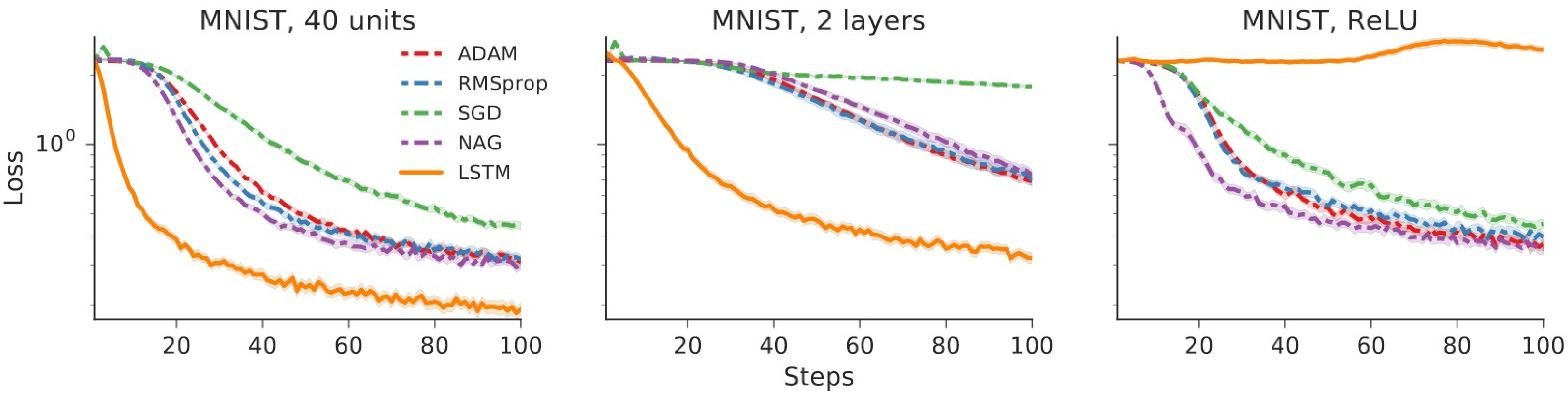
[4] Andrychowicz et al. 2016

Learning to learn by gradient descent by gradient descent



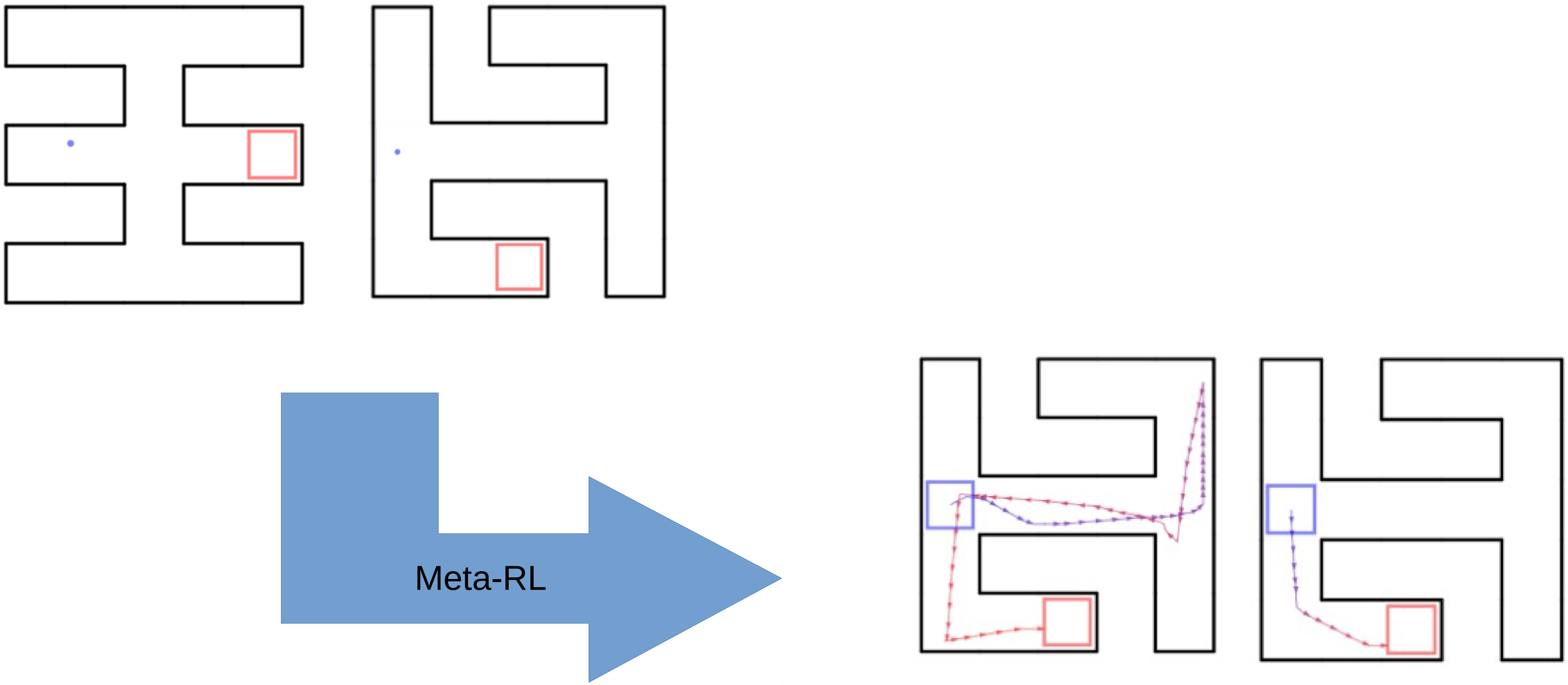
[4] Andrychowicz et al. 2016

Learning to learn by gradient descent by gradient descent



[4] Andrychowicz et al. 2016

Meta-Learning in RL



[5] Duan et al. 2016

Meta-Learning in RL

Reinforcement learning:

$$\theta^\star = \arg\max_\theta E_{\pi_\theta(\tau)}[R(\tau)]$$

$$= f_{\mathrm{RL}}(\mathcal{M}) \qquad \mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$$

MDP

Meta-reinforcement learning:

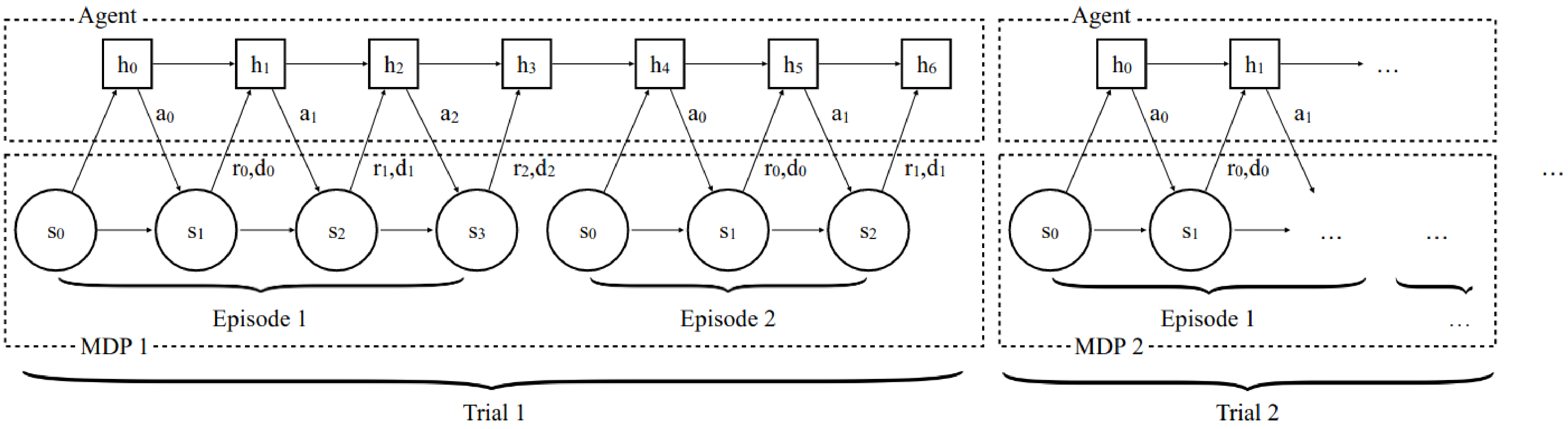
$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_\theta(\mathcal{M}_i)$$

MDP for task i

[5] Duan et al. 2016

RL² – Fast RL via Slow RL

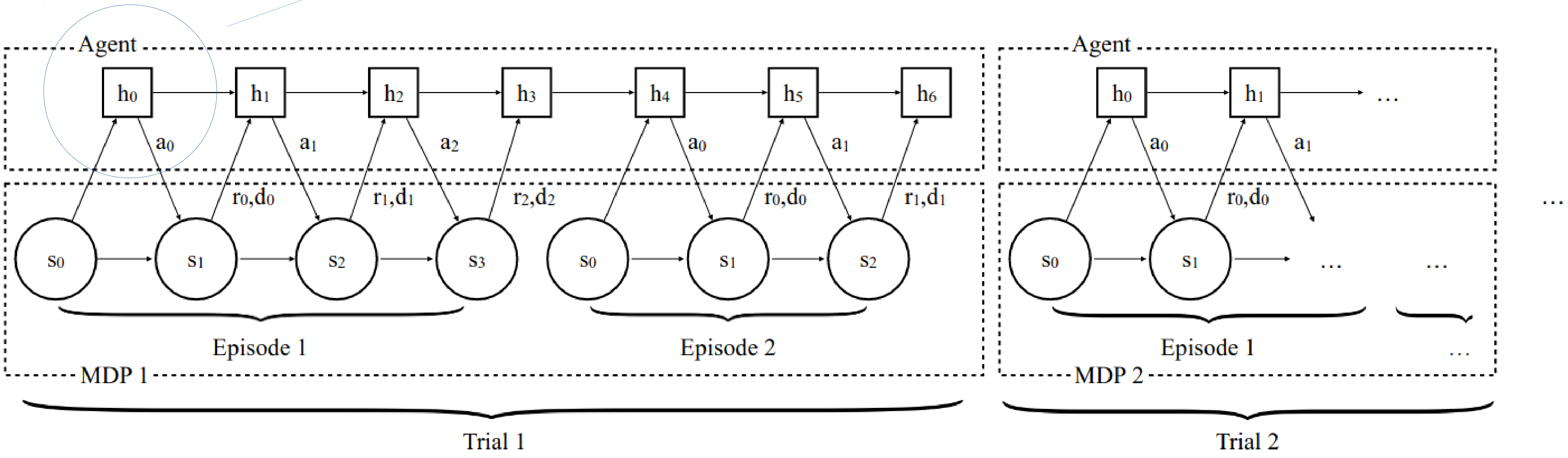


"We view the learning process of the agent itself as an objective, which can be optimized using standard RL algorithms."

[5] Duan et al. 2016

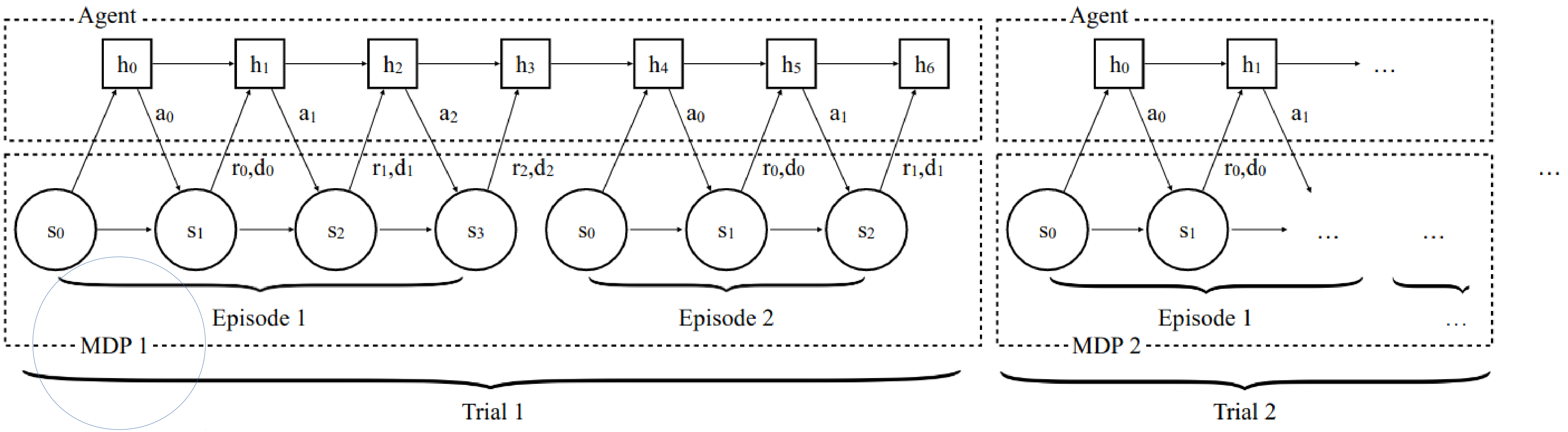
RL² – Fast RL via Slow RL

Policy is modeled by a RNN



[5] Duan et al. 2016

RL² – Fast RL via Slow RL

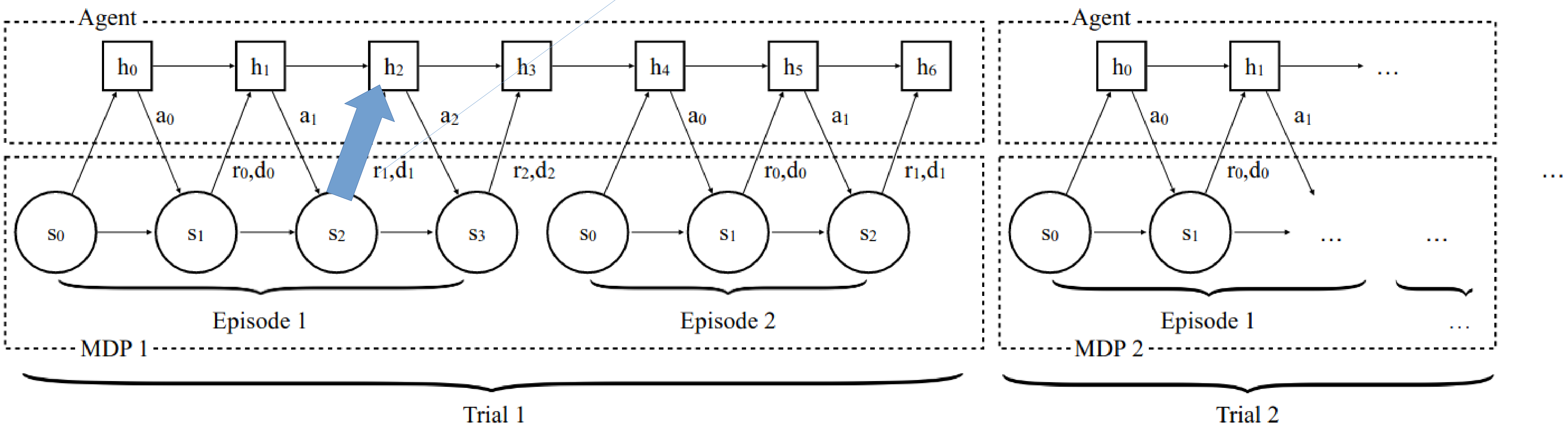


Environment is modeled by a MDP

[5] Duan et al. 2016

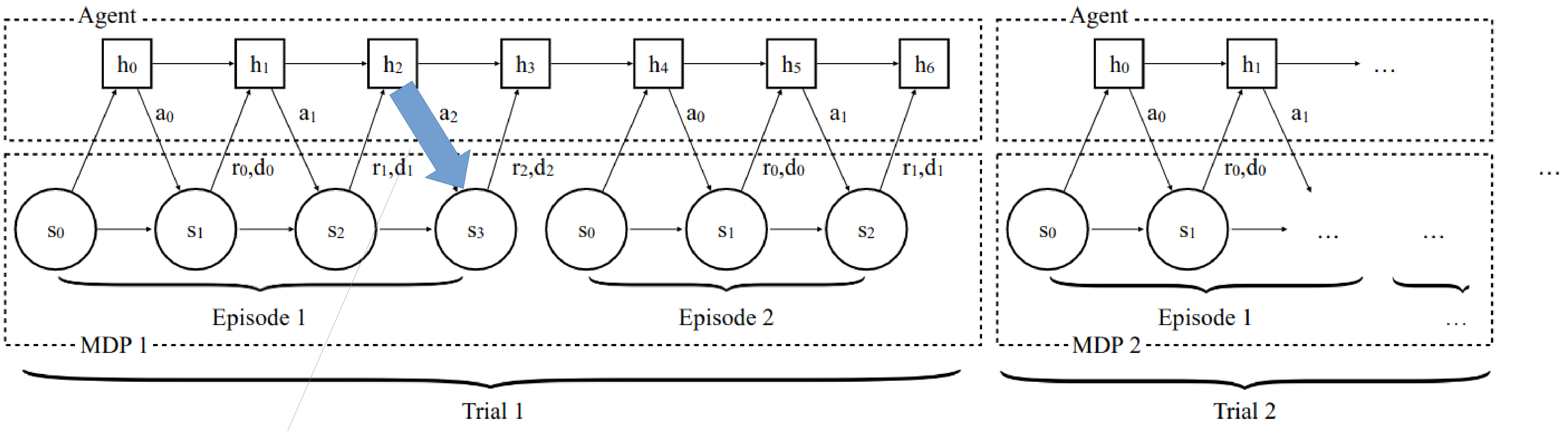
RL² – Fast RL via Slow RL

next state, action, reward and termination flag



[5] Duan et al. 2016

RL² – Fast RL via Slow RL

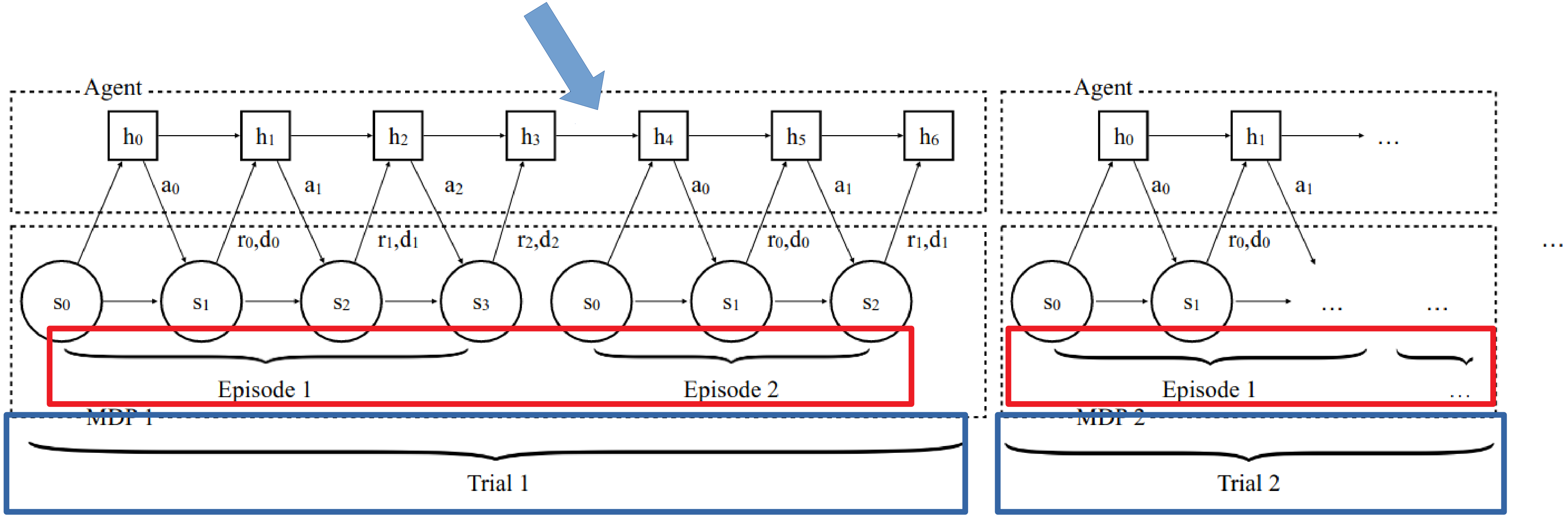


action

[5] Duan et al. 2016

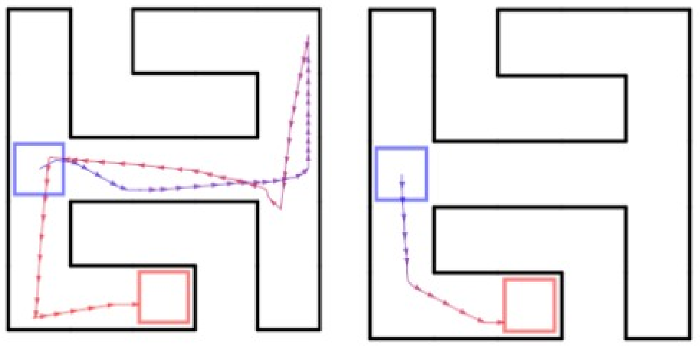
RL² – Fast RL via Slow RL

Hidden state is kept

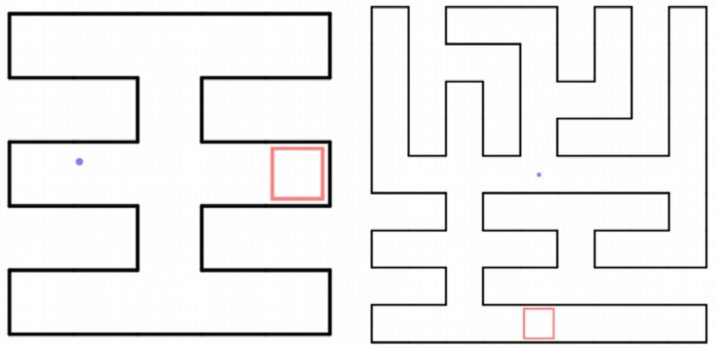


[5] Duan et al. 2016

RL² – Fast RL via Slow RL



Second trajectory is almost always shorter



Generalizes to larger mazes

[5] Duan et al. 2016

Thought Experiment

We assumed that learning optimization algorithms was better than hand-designing optimization algorithms. But why do we think that hand-designing meta-learning algorithms is optimal and why don't we meta-meta-learn them?

METACEPTION

References

[1] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[3] Aniruddh Raghu et al. "Rapid learning or feature reuse? towards understanding the effectiveness of maml". In: *arXiv preprint arXiv:1909.09157* (2019).

[4] Marcin Andrychowicz et al. "Learning to learn by gradient descent by gradient descent". In: *Advances in neural information processing systems*. 2016, pp. 3981–3989.

[5] Yan Duan et al. "Rl ˆ2: Fast reinforcement learning via slow reinforcement learning". In: *arXiv preprint arXiv:1611.02779* (2016).